ExtremeAnalytics® Deployment Guide

05/2021
9037060-00
Subject to Change Without Notice

Copyright © 2021 Extreme Networks, Inc. All Rights Reserved.

Legal Notices

Extreme Networks, Inc., on behalf of or through its wholly-owned subsidiary, Enterasys Networks, Inc., reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks/

Contact

If you require assistance, contact Extreme Networks using one of the following methods.

- Global Technical Assistance Center (GTAC) for Immediate Support
  - **Phone:** 1-800-998-2408 (toll-free in U.S. and Canada) or 1-603-952-5000. For the Extreme Networks support phone number in your country, visit: www.extremenetworks.com/support/contact
  - **Email:** support@extremenetworks.com. To expedite your message, enter the product name or model number in the subject line.

- GTAC Knowledge — Get on-demand and tested resolutions from the GTAC Knowledgebase, or create a help case if you need more guidance.
- The Hub — A forum for Extreme customers to connect with one another, get questions answered, share ideas and feedback, and get problems solved. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
- Support Portal — Manage cases, downloads, service contracts, product licensing, and training and certifications.

Table of Contents

About This Guide

This document describes how to design your ExtremeAnalytics solution deployment.

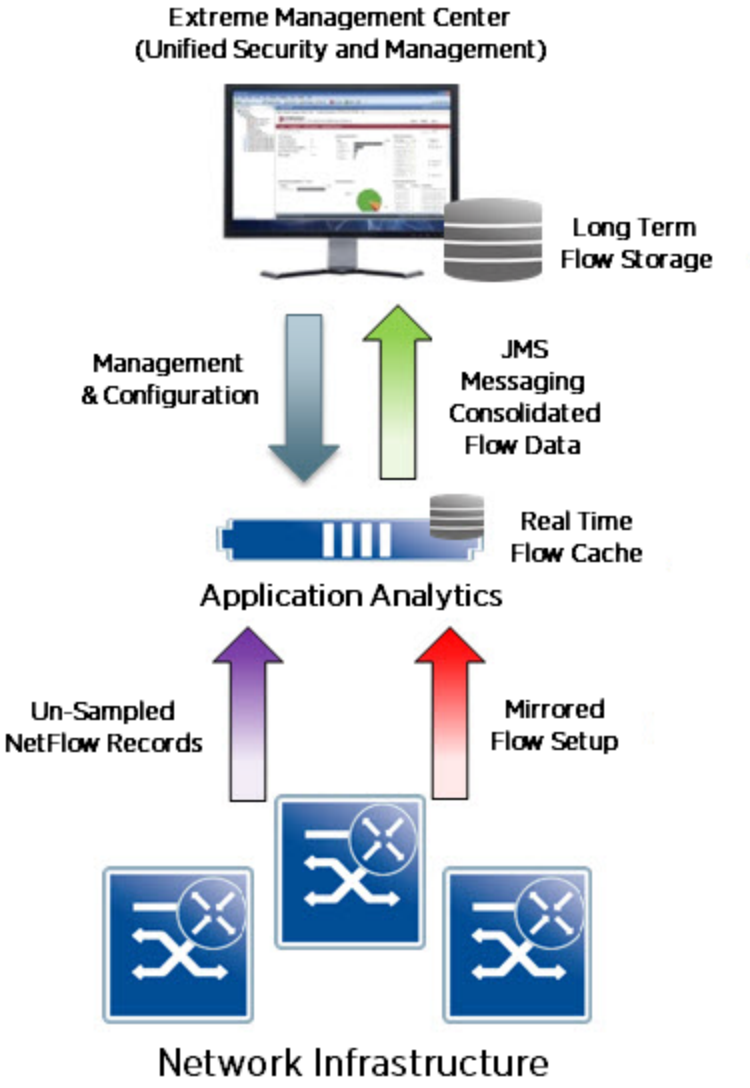
This document is intended for experienced network administrators who are responsible for implementing and maintaining communications networks.

Introduction

This section provides an overview and lists the requirements for your ExtremeAnalytics deployment.

Deployment Overview

The following figure shows a simplified architecture and information flows in an ExtremeAnalytics deployment.



ExtremeAnalytics Deployment

An Extreme S- or K-Series CoreFlow2 switch forwards the following to an ExtremeAnalytics engine:

- Unsampled NetFlow, which provides an accurate statistical representation of all flows mirrored for application identification.
- The first 15 packets of each flow, via a forensic policy mirror.

  The mirrored traffic can be delivered to the ExtremeAnalytics engine by two methods:

  - A local traffic mirror
  - Remote mirroring through GRE (Generic Routing Encapsulation) L2 tunneling. Through remote mirroring, a single ExtremeAnalytics engine can receive traffic feeds from multiple switches in the network without being directly connected to them.

On the ExtremeAnalytics engine, the application stream is assembled and fingerprints are applied to identify the applications. This information is then combined with the corresponding NetFlow record. The combined record, which provides IP and TCP/UDP information, application name and category, TCP and application response times, and application metadata, is sent to the ExtremeCloud™ IQ - Site Engine server for graphing and storage.

Deployment Requirements

Deploying an ExtremeAnalytics solution requires the following:

- Extreme S- or K-Series CoreFlow2 switches
- ExtremeAnalytics engine — Available as a hardware engine or a virtual engine.

  For deployments requiring more than 10G monitored bandwidth, you can install the PV-A-300-10G-UG I/O module in the ExtremeAnalytics engine to enhance the interface bandwidth. Refer to *ExtremeAnalytics PV-A-305 Installation Guide* for information on installing a 10G interface.
  Note: Virtual engines are bandwidth-bound by the underlying host interface bandwidth. If the host interface can operate at 10Gbps, the virtual interface can be reassigned to a new hardware interface without any change in the ExtremeAnalytics engine.
- ExtremeCloud IQ - Site Engine management—Besides the configuration and monitoring of the ExtremeAnalytics solution, provides the long term storage and

reporting, presenting the correlated data with contextual information. Additionally, ExtremeCloud IQ - Site Engine provides that data to other IT systems via the Connect tab.

- Licenses:
  - NMS-ADV (ExtremeCloud IQ - Site Engine Advanced License) — ExtremeAnalytics management and configuration requires an NMS-ADV license.
  - PV-FPM (ExtremeAnalytics Flow License)—Enables the ExtremeAnalytics flow processing capability on the ExtremeAnalytics engine.
  - Feature-specific licenses—CoreFlow2 features such as GRE tunneling may require a specific license depending on the version of firmware running in your CoreFlow2 switch.

Designing Your ExtremeAnalytics Deployment

To design a reliable and accurate ExtremeAnalytics deployment, consider the following:

Traffic Domains

To ensure that an ExtremeAnalytics engine does not receive duplicate traffic, split your network into non-overlapping functional or topological traffic domains. Assign one traffic domain to each ExtremeAnalytics engine in your deployment.

Delivering traffic from only one traffic domain provides each ExtremeAnalytics engine with an accurate representation of the traffic in the domain and keeps the mirrored traffic within the scalability limits of the engine.

Use the following guidelines when creating your traffic domains:

- Physical network layers: Edge, distribution, core.
- IP network boundaries, such as Internet, intranet, and DMZ. These network boundaries are easily identified and provide easily monitored points for the ExtremeAnalytics engine.
- Functional domains, such as sales and R&D. Isolating traffic from one functional domain or another can be difficult. To simplify your deployment, you should identify a single port for each functional domain to use for traffic monitoring.
- NAT boundaries: Use NAT boundaries as ExtremeAnalytics traffic domain boundaries. A NAT section inside a traffic domain can lead to unexpected results because the ExtremeAnalytics engine will not be able to identify the original flow

and the NAT flow. If a traffic domain contains the original flows and the NAT flows, the flows will be counted twice for the two different source addresses.

Monitored Points

You must establish monitored points in the CoreFlow2 switch to provide a reliable and accurate traffic sample to an ExtremeAnalytics engine. In a reliable and accurate traffic sample, the statistics obtained are the same statistics that would be obtained from all the traffic in the domain. Ensure that your traffic samples meet the following requirements:

- Every flow in the traffic domain must be represented in the sample. To do this, you must carefully plan the points where the traffic is sampled or mirrored.
- Every flow in the traffic domain must appear only once in the sample. Your deployment must avoid traffic samples that contain multiple copies of the same flow.

To minimize the collection of unidirectional or duplicate flows:

- Map all ingress (RX) pathways where network traffic can enter into a particular traffic domain.
    - The ingress pathways can traverse multiple switches and therefore will usually require a detailed and accurate network diagram to pinpoint them.
    - If any ingress ports to the traffic domain are left out of the deployed configuration, the result will be inaccurately tagged unidirectional flows.
- Ensure that any multi-pathed traffic is delivered to the same ExtremeAnalytics engine.
- Do not include intra-traffic domain pathways built for redundancy between switches as these links will lead to duplicate flows.
- Configure the ingress network pathways isolated during the planning process to capture only the inbound traffic to the port. For each switch involved in a traffic domain, ensure that NetFlow and the ExtremeAnalytics traffic mirror on the isolated ports are enabled in the RX direction only.

For scenarios that can lead to traffic capture issues, see Common Flow Collection Issues.

You can establish your monitored points at various locations in your network:

- Edge
- Distribution
- Core
- DMZ

Edge

Extreme CoreFlow2 switches can provide NetFlow statistics and traffic mirroring capabilities on all edge ports. By deploying all uni-directional mirrors, all traffic flows are captured only when entering the network. Assuming that all edge traffic is delivered to another edge point, this strategy ensures a complete traffic sample. If edge devices exchange traffic with devices in other areas of the network, such as the data center, DMZ, or internet, you can ensure that all traffic is captured by adding monitored points in other network layers.

Distribution

In terms of configuration and scalability, monitoring all traffic in the edge layer can be difficult. Mirroring at the distribution layer provides a similar level of accuracy and, because it requires fewer mirroring configurations, greater scalability. The risk with this strategy is the loss of traffic switched locally at the edge.

Core

If you expect highly centralized traffic flows, such as edge to data center or edge to Internet, establishing monitored points at the core level offers the greatest visibility with the fewest configuration requirements.

DMZ

The DMZ traffic domain is an independent network area, usually a set of VLANs or even network devices that connect with the rest of the network through firewalls. These firewall interfaces can provide the required traffic sample for all ingress and egress traffic. You should assume that most of the traffic in the DMZ will be with systems and users outside of the DMZ. If you expect a large amount

of intra-DMZ traffic, you must adapt the traffic sampling strategy to account for this and replicate the strategy used in an edge or core deployment.

Common Flow Collection Issues

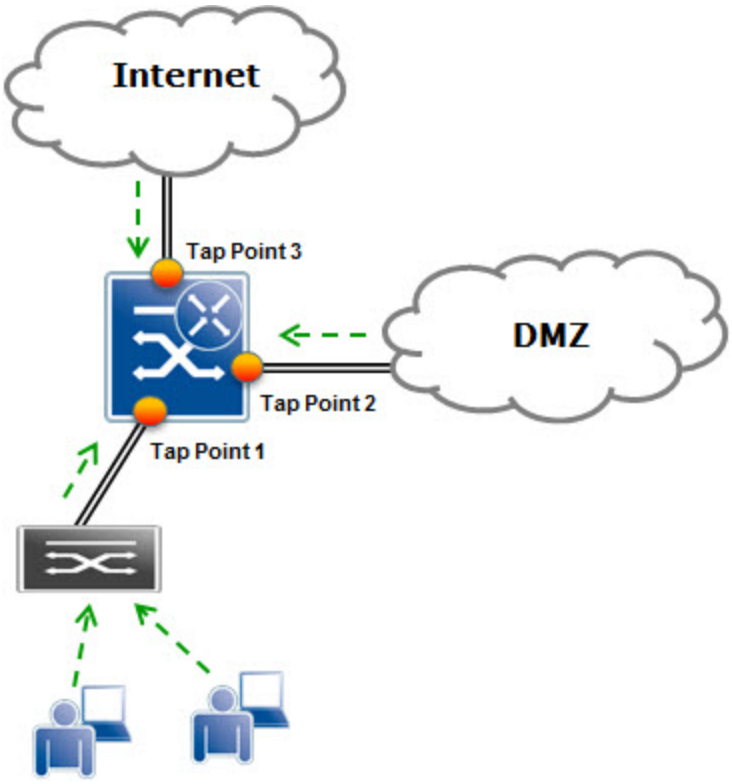
Ensure that your ExtremeAnalytics deployment accounts for the following potential problem areas:

- Unidirectional Flows
- Duplicate Flows
- Asymmetric Routing
- Network Load Balancing

Unidirectional Flows

By default, the Extreme CoreFlow2 switches forward NetFlow and the application traffic mirror for all ingress flows. This configuration can cause unidirectional flows if not deployed properly.

The following figure shows three tap points configured on a switch. When a user accesses the Internet, the traffic is captured by Tap Point 1 on the way out to the Internet and at Tap Point 3 for the return traffic. Include both tap points in the traffic domain.

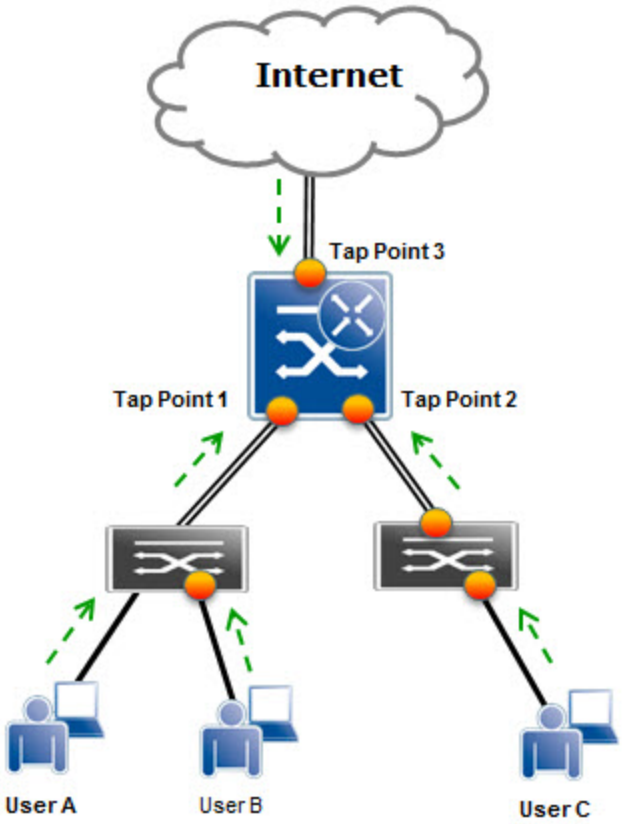


Unidirectional Flows

Duplicate Flows

Duplicate flows occur when the same flow is seen at multiple tap points included in an ExtremeAnalytics traffic domain.

In the following figure, User A's traffic, which is logged once on each path to and from the Internet, is counted correctly. User B's traffic is seen twice on the way to the Internet and once on the return path from the Internet. User C's traffic is logged twice in both directions. To avoid duplicate flows, limit the collection locations or create multiple traffic domains to keep the duplicate information separated.



Duplicate Flows

Asymmetric Routing

Asymmetric routing occurs when a packet takes a path from source A to destination B but then the return packet from B takes a different path back to A. Because there are no perceived performance issues associated with asymmetric routing for normal applications traveling across the network, network administrators are often unaware that asymmetric routing is occurring. The issue becomes evident once NetFlow or a port mirror is enabled for a specific path and only one half of a TCP conversation is seen. This causes ExtremeAnalytics to produce erroneous results.

Network Load Balancing

If your network includes a load balancing configuration, ensure that all necessary links are covered by the ExtremeAnalytics solution. If covering all links is impossible because of physical constraints or possible flow duplication, your

ExtremeAnalytics deployment may require collection at specific network choke points.

CoreFlow2 Switch Deployment

You can deploy a CoreFlow 2 switch for the ExtremeAnalytics solution in two ways:

- In-line—If you have CoreFlow2 switches in your network, the NetFlow and traffic mirroring capabilities can be provided by the network itself. For additional information, see In-Line Deployment.
- Overlay—If you do not currently have CoreFlow2 switches installed in your network, you can install CoreFlow2 switches as overlays to provide the required NetFlow and traffic mirroring capabilities. For additional information, see Overlay Deployment.

In-Line Deployment

Extreme S- and K-Series CoreFlow2 switches can capture traffic directly traversing them and sample the flow to provide the ExtremeAnalytics engine with an accurate representation of the application data.

Design your flow mirror configuration according to the guidelines in Traffic Domains and Monitored Points to ensure the domain does not include duplicate flows.

The best practices include the following:

- Configure unidirectional port mirrors (RX or TX only).
- Deploy the monitored ports in such a way that all potential egress or ingress ports in a domain are mirrored.

RX, or ingress, is the default for policy mirrors and NetFlow configurations. Unless you require egress configurations, ingress configurations are highly recommended.

For full visibility, an in-line flow collection deployment should have CoreFlow2 switches deployed across the edge, distribution, core, data center, and/or DMZ of the network. You can use an already deployed CoreFlow2 switch infrastructure or deploy CoreFlow2 switches at one layer of the network to collect application tagged flows from all of the previously defined traffic domains.

To ensure a smooth migration and integration of non-intelligent network layers and segments, you can start by enabling the ExtremeAnalytics solution for just one or two layers, such as data center and distribution.

Overlay Deployment

An overlay flow collection deployment is similar to an in-line deployment in that flow collection can take place across the edge, distribution, core, data center, and DMZ. An overlay deployment is appropriate if your network consists of network switches that lack the capabilities of CoreFlow2 switches to provide mirror traffic feed and unsampled NetFlow statistics to the ExtremeAnalytics engine. You can use a passive network tap to direct traffic to an out-of-band CoreFlow2 switch that generates the required unsampled NetFlow stats and traffic mirror.

A full mirror of the desired traffic is pulled from the existing network infrastructure using mirroring, span, or passive tap solutions and is forwarded into a CoreFlow2 switch, which then performs the ExtremeAnalytics flow and packet processing on the forwarded traffic.

For a complete ExtremeAnalytics overlay mode deployment, each defined traffic domain requires the same network pathway planning and traffic collection to ensure that you have selected the correct choke point ports for ExtremeAnalytics monitoring.

Note: For a successful overlay mode deployment, your network infrastructure must be capable of a bi-directional line rate mirror for the designated choke point ports.

ExtremeAnalytics Engine Deployment

The ExtremeAnalytics engine supports multiple deployment modes to suit different network environments and connectivity characteristics:

- Single Interface—A single interface is used for both management and monitoring traffic. You must configure a GRE tunnel for traffic monitoring.
- Dual Interface Mirrored—Separate interfaces are configured for management and monitoring traffic. The monitoring interface will be put into tap mode for traffic monitoring.

- Dual Interface Tunnel Mirrored—Separate interfaces are configured for management and monitoring traffic. The monitoring interface will get its own IP address. You must configure GRE tunnels for traffic monitoring.

For more information on configuring the ExtremeAnalytics engine deployment mode, see the *Installation Guide* for your ExtremeAnalytics engine.

Traffic Mirror Forwarding Methods

Use one of the following methods to deliver the traffic mirror from a CoreFlow2 switch to an ExtremeAnalytics engine:

- A direct physical connection from the CoreFlow2 destination mirror port(s) to an Ethernet interface on the ExtremeAnalytics engine.
- A GRE tunnel:
  - For remote networks, GRE tunnels provide an affordable solution to transfer the mirror traffic to a centralized ExtremeAnalytics engine, eliminating the need for individual ExtremeAnalytics engines at the remote locations.
  - For data center deployments, GRE tunnels reduce the number of Ethernet monitoring interfaces required on an ExtremeAnalytics engine, which can be especially critical for ExtremeAnalytics deployments built upon VMware.

  Using a GRE tunnel, each CoreFlow2 switch wraps the ExtremeAnalytics mirror into a GRE tunnel that terminates either inside the ExtremeAnalytics engine or, for scalability reasons, on a CoreFlow2 switch in front of the ExtremeAnalytics engine. Once the GRE header is removed from the ExtremeAnalytics mirror traffic, the ExtremeAnalytics processing continues as if the mirror was physically connected.

  If you use GRE tunnels, ensure the following:

  - Traffic delay and bandwidth availability along the path of the mirroring tunnels do not impair the capability of the ExtremeAnalytics engine to calculate application statistics.
  - Jumbo Ethernet frame functionality is enabled.

  You must configure the GRE tunnels on both the CoreFlow2 switch and the ExtremeAnalytics engine. For an example of configuring a GRE tunnel on a CoreFlow2 switch, see Sample CoreFlow2 Switch Configurations. For

information on configuring GRE tunnels in an ExtremeAnalytics engine, see the *Installation Guide* for your ExtremeAnalytics engine.

Note: Do not use WAN links to transport mirrored traffic to the ExtremeAnalytics engine.

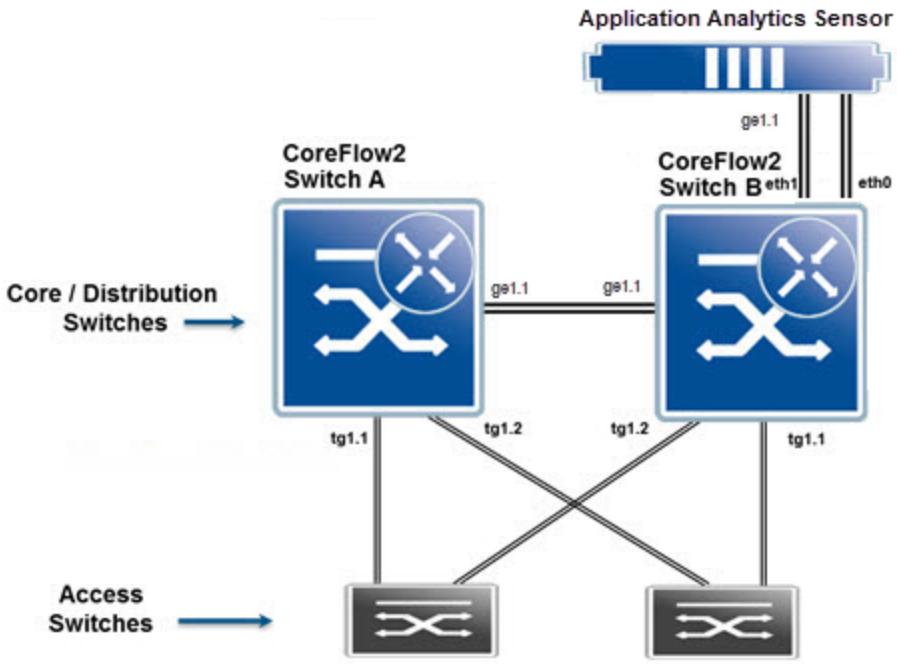
Sample CoreFlow2 Switch Configurations

The following section contains sample configurations for CoreFlow2 switches:

- In-Line (ToR Switches)
- In-Line (Bonded CoreFlow2 Switch)
- Overlay Mode

The sample configurations assume that you have configured the ExtremeAnalytics engine with the corresponding interfaces and GRE tunnels. For more information, see the *Installation Guide* for your ExtremeAnalytics engine.

In-Line (ToR Switches)



The servers in this network are connected to the ToR switches and must traverse core switches Switch A and Switch B to reach their clients. It is assumed that spanning tree will keep a loop-free topology and no load traffic balancing

algorithms are run in the interfaces between the ToR/EoR switches and Switch A and Switch B.

In this example, the traffic inside the ToR switches is considered irrelevant as the deployment focuses on the traffic going through core switches Switch A and B. Because Switch A needs to traverse Switch B to reach the ExtremeAnalytics engine, Switch A will use a GRE tunnel to direct its mirror to port eth1 of the engine.

The setup of a GRE tunnel in CoreFlow2 switches requires an unused physical port assigned as an endpoint of the GRE tunnel. Port ge.1.3 is used in Switch A.

- Switch A management IP: 192.168.10.11/24
- Switch B management IP: 192.168.10.12/24
- ExtremeAnalytics engine eth0: 192.168.20.100/24
- ExtremeAnalytics engine eth1: 192.168.30.100/24

GRE tunnels can be started from any interface in the origin switch. To avoid management and interface availability issues, use loopback interfaces as originators of the GRE traffic in Switch A and Switch B. The state of the loopback interfaces is not related to the egress state of VLANs in the switch so these interfaces are always up and available. You must set up routing appropriately so that the loopback interface can reach eth1 of the ExtremeAnalytics engine and the default gateway of the ExtremeAnalytics engine can reach the loopback interface.

Switch A Configuration

To configure switch A:

1. Enter the following commands to ensure that ports used for GRE tunnels are statically configured for speed and duplex:

   ```
   set port duplex ge.1.1-3 full

   set port speed ge.1.1-3 1000
   ```

2. Enter the following commands to enable NetFlow reporting in the monitored port and export its data to the NetFlow receiver (eth1 interface) in the ExtremeAnalytics engine:

   ```
   set netflow export-interval 1
   ```

```
set netflow export-destination 192.168.30.100 2055

set netflow export-version 9

set netflow export-data enable mac

set netflow port tg.1.1-2 enable rx

set netflow template refresh-rate 30 timeout 1

set netflow cache enable
```

3. Enter the following commands to create the policy mirror that samples the traffic flows. This mirror delivers only relevant application information to the ExtremeAnalytics engine:

```
set mirror create 1

set mirror 1 mirrorN 15

set mirror ports ge.1.3 1
```

4. Enter the following commands to create a policy using this mirror. In this case, the traffic continues flowing in the switch, so a PVID of 4095 is configured in the policy:

```
set policy profile 1 name Application pvid-status enable
pvid 4095 mirror-destination 1

set policy rule admin-profile port tg.1.1-2 mask 16 port-
string tg.1.1-2 admin-pid 1
```

5. Enter the following commands from the routing configuration terminal in the CoreFlow2 switch to create a GRE tunnel to transport this mirror to the eth1 interface of the ExtremeAnalytics engine. This example assumes that the IP address 10.10.10.1 is routable in the network:

```
interface loop.0.1

ip address 10.10.10.1 255.255.255.255 primary

no shutdown

exit

interface tun.0.1

tunnel destination 192.168.30.100
```

```
tunnel mode gre l2 ge.1.3

tunnel mirror enable

tunnel source 10.10.10.1

no shutdown

exit
```

6. Enter the following command to enable jumbo frames on the physical interfaces where GRE traffic will traverse:
   `set port jumbo enable ge.1.1`

Switch A is now configured.

Switch B Configuration

To configure switch B:

1. Enter the following commands to ensure that ports used for GRE tunnels are statically configured for speed and duplex:

   ```
   set port duplex ge.1.1-3 full

   set port speed ge.1.1-3 1000
   ```

2. Enter the following commands to enable NetFlow reporting in the monitored port and export its data to the NetFlow receiver (eth1 interface) in the ExtremeAnalytics engine:

   ```
   set netflow export-interval 1

   set netflow export-destination 192.168.30.100 2055

   set netflow export-version 9

   set netflow port tg.1.1-2 enable rx

   set netflow template refresh-rate 30 timeout 1

   set netflow cache enable
   ```

3. Enter the following commands to create the policy mirror:

   ```
   set mirror create 1

   set mirror 1 mirrorN 15
   ```

```
set mirror ports ge.1.3 1
```

4. Enter the following commands to create a policy using this mirror. In this case, the traffic continues flowing in the switch, so a PVID of 4095 is configured in the policy:

```
set policy profile 1 name Application pvid-status enable
pvid 4095 mirror-destination 1

set policy rule admin-profile port tg.1.1-2 mask 16 port-
string tg.1.1-2 admin-pid 1
```

5. Enter the following commands from the routing configuration terminal in the CoreFlow2 switch to create a GRE tunnel to transport this mirror to the eth1 interface of the ExtremeAnalytics engine. This example assumes that the IP address 10.10.10.2 is routable in the network:

```
interface loop.0.1

ip address 10.10.10.1 255.255.255.255 primary

no shutdown

exit

interface tun.0.1

tunnel destination 192.168.30.100

tunnel mode gre l2 ge.1.3

tunnel mirror enable

tunnel source 10.10.10.2

no shutdown

exit
```

6. Enter the following command to enable jumbo frames on the physical interfaces where GRE traffic will traverse:

```
set port jumbo enable ge.1.1
```

Switch B is now configured.

In-Line (Bonded CoreFlow2 Switch)

In in-line mode, if Switches A and B are two bonded CoreFlow2 switches, the ExtremeAnalytics engine is directly connected to the bonding so there is no need to use a GRE tunnel to forward mirrored traffic from Switch A to the ExtremeAnalytics engine. In addition, a 10 gigabit port is used for connectivity from Switch B to the ExtremeAnalytics engine.

Assuming that both switches are S4 switches, Switch A is chassis 1 in the bonded pair, and Switch B is chassis 2, the port numbering is as follows:

In-Line Mode Virtual Switch Bonding Port Numbers

Switch	Switch port number	VSB port number
A	tg.1.1	tg.1.1
	tg.1.2	tg.1.2
B	tg.1.1	tg.5.1
	tg.1.2	tg.5.2
	tg.1.3	tg.5.3

Connectivity remains the same. In a redundant bonded pair, ports tg.1.1 and tg.1.2 in both switches are aggregated as a lag. Assuming that lag.0.1 and lag.0.2 are created in the bonded pair with the following assignment, port numbering follows the bonded pair numbering convention:

- lag.0.1 -> tg.1.1 tg.5.1
- lag.0.2 -> tg.1.2 te.5.2

In-Line (Bonded CoreFlow2 Switch) Configuration

The configuration in this example is the same as the configuration presented in In-Line (ToR Switches) with the following changes:

- VSB port numbering and LAGs are used.
- A GRE tunnel is not created.

To configure the switch:

1. Enter the following commands to enable NetFlow reporting in the monitored port and export its data to the NetFlow receiver (eth1 interface) in the ExtremeAnalytics engine. In this case the monitored ports are the LAGs:

```
set netflow export-interval 1

set netflow export-destination 192.168.30.100 2055

set netflow export-version 9

set netflow port lag.0.1-2 enable rx

set netflow template refresh-rate 30 timeout 1

set netflow cache enable
```

2. Enter the following commands to create the policy mirror:

```
set mirror create 1

set mirror 1 mirrorN 15

set mirror ports tg.5.3 1
```

3. Enter the following commands to create a policy using this mirror. In this case, the traffic continues flowing in the switch, so a PVID of 4095 is configured in the policy:

```
set policy profile 1 name Application pvid-status enable
pvid 4095 mirror-destination 1

set policy rule admin-profile port lag.0.1 mask 16 port-
string lag.0.1 admin-pid 1

set policy rule admin-profile port lag.0.2 mask 16 port-
string lag.0.2 admin-pid 1
```

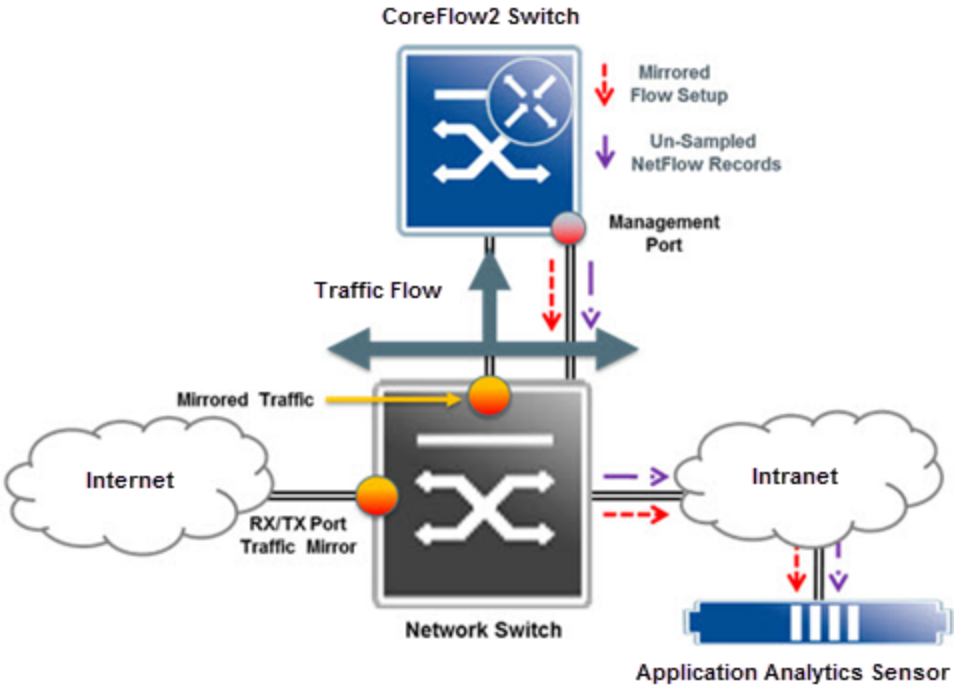
   Note: If you plan to use GRE tunneling as part of this setup to mirror traffic to a remote location, VSB only allows GRE configuration if dedicated bonding ports are in use.

The switch is now configured.

Overlay Mode

The overlay mode configuration is generally used at a choke point of a network consisting of third-party gear or Extreme switches that do not support the policy mirroring (mirrorN) capabilities of the CoreFlow2 switches. In this example configuration, a bidirectional port mirror is created for all traffic coming from or going to the Internet. To filter the data from that mirror and generate the NetFlow records, an overlay SSA is used. Because this is a situation with less

traffic, a single interface is used on the ExtremeAnalytics engine to converge all policy mirrored traffic, all NetFlow, and all management traffic. A GRE tunnel is used to forward the mirror traffic from the SSA to the ExtremeAnalytics engine. Port ge.1.5 is used on the SSA overlay switch for management and GRE tunneling.



Overlay Mode Switch Configuration

This example uses the following IP addresses:

- CoreFlow2 switch management IP address: 192.168.10.11
- ExtremeAnalytics engine eth0 IP address: 192.168.30.100

To configure the switch:

1. Enter the following commands to ensure that port ge.1.24, which is used for GRE tunneling, is statically configured for speed and duplex. Ensure that spanning tree configuration keeps ports ge.1.3, ge.1.24 and ge.1.5 loop free and forwarding traffic:

   ```
   set port duplex ge.1.24 full

   set port speed ge.1.24 1000
   ```

```
set spantree portadmin ge.1.3 disable
```

2. Enter the following commands to enable NetFlow reporting for the monitored port and export its data to the NetFlow receiver (eth0 interface) in the ExtremeAnalytics engine:

```
set netflow export-interval 1

 set netflow export-destination 192.168.30.100 2055

set netflow export-version 9

set netflow port ge.1.3 enable rx

set netflow template refresh-rate 30 timeout 1

set netflow cache enable
```

3. Enter the following commands to create the policy mirror:

```
set mirror create 1

set mirror 1 mirrorN 15

set mirror ports ge.1.24 1
```

4. Enter the following commands to create a policy using this mirror. In this case, the traffic should be dropped after the switch forwards the relevant frames to the ExtremeAnalytics engine, so a PVID of 0 is configured in the policy:

```
set policy profile 1 name Application pvid-status enable
pvid 0 mirror-destination 1

set policy rule admin-profile port ge.1.3 mask 16 port-
string ge.1.3 admin-pid 1
```

5. Enter the following commands to create a GRE tunnel to transport this mirror to the ExtremeAnalytics engine. In this example the GRE tunnel follows the path through the management interface connecting the CoreFlow2 switch to the network for management. Ensure that proper routing exists to reach the ExtremeAnalytics engine from the management interface of the overlay SSA.

```
interface loop.0.1

ip address 10.10.10.1 255.255.255.255 primary

no shutdown
```

```
exit
interface tun.0.1
tunnel destination 192.168.30.100
tunnel mode gre l2 ge.1.24
tunnel mirror enable
tunnel source 10.10.10.1
no shutdown
exit
```

6. Enable jumbo frames on the physical interfaces through which GRE traffic will traverse. You must enable jumbo frames on each interface that the GRE traffic traverses across the entire network.

   ```
   set port jumbo enable ge.1.5
   ```

The switch is now configured.

Alternative Designs

You can adapt the ExtremeAnalytics solution to your network requirements. This section describes deployment changes for the following scenarios:

- More than Two Ethernet Interfaces
- More than Four GRE Tunnels
- VMWare TAP Interface

Note: Use these designs only if your deployment requirements are not met by the designs previously described in this guide.

More than Two Ethernet Interfaces

In cases of specific bandwidth restraints or design requirements, you may need more than two Ethernet interfaces. In examples in Unidentified Entries in ExtremeAnalytics, the mirrorN feature of CoreFlow2 switches redirected only the first 15 packets of each flow to the ExtremeAnalytics engine. This reduces the traffic managed by the engine to under 1Gbps for most deployments. If, however, your total bandwidth is still greater than 1Gbps, you can enable more Ethernet interfaces for traffic monitoring in the engine.

1. Add the following lines to the /etc/network/interfaces file on the ExtremeAnalytics engine for each additional interface. In the example below, eth2 is being added:

   ```
   auto eth2
   iface eth2 inet manual
       up ifconfig eth2 0.0.0.0 up
       up ip link set eth2 promisc on
       down ip link set eth2 promisc off
       down ifconfig eth2 down
   ```

   If the additional Ethernet interfaces are used to terminate GRE tunnels, additional commands must be added. In the example below eth2 will be used as a GRE interface 2 with the IP address of 192.168.40.100:

   ```
   # The secondary network interface
   auto eth2
   ```

```
iface eth2 inet manual
    up ifconfig eth2 192.168.40.100 up
    up ip link set eth2 promisc on
    down ip link set eth2 promisc off
    down ifconfig eth2 down


#GRE tap interfaces
auto gre2
iface gre2 inet static
    address 0.0.0.0
    pre-up ip  link add gre2 type gretap remote $SWITCH_IP
local 192.168.40.100 ttl 255
    post-down ip link del gre2
```

The newly created interfaces can be used as NetFlow destinations or GRE traffic endpoints in the switches' configurations.

2. Restart the engine to enable the newly created interfaces.
3. Create the `/opt/appid/conf/appidconfig-local.xml` file, if necessary.
4. Add the following lines to the `/opt/appid/conf/appidconfig-local.xml` file to declare the interfaces to the ExtremeAnanlytics engine to use them as monitoring interfaces:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
  <Interfaces>
    <Interface name="eth1"/>
    <Interface name="eth2"/> /*one line per each newly
created interface
  </Interfaces>
</Configuration>
```

Note: If the file exists and contains the <Configuration> tag, add the interface lines between the <Configuration></Configuration> tags.

```
 <Interface name="eth2"/>
    <Interface name="eth3"/> /*one line per each newly
created interface
```

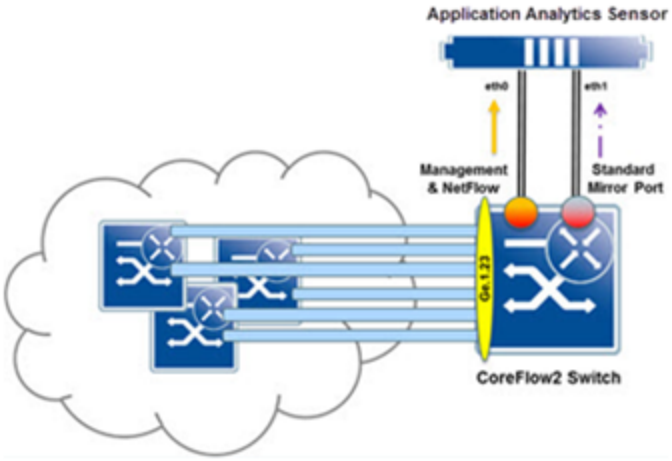
5. Restart the server process for the new configuration to take effect using the following command:

```
appidctl restart
```

Note: Virtual ExtremeAnanlytics engines are bandwidth-bound by the underlying host interface bandwidth. They can grow up to the bandwidth that the virtualization platform can offer in a virtual interface.

More than Four GRE Tunnels

ExtremeAnalytics engines can support up to four GRE tunnels. If your deployment requires more than four GRE tunnels, the GRE tunnels can be terminated in a CoreFlow2 switch and the traffic can be delivered to the ExtremeAnanlytics engine using standard traffic mirroring, as shown in the following figure.



.

More than Four GRE Tunnels

In this scenario, the CoreFlow2 switch is terminating six tunnels from other switches and delivering the content of the tunnels in port ge.1.23.

The tunnels terminate at the same loopback IP address in the CoreFlow2 switch, 10.10.10.10. The other switches have addresses 10.10.10.1 through 10.10.10.5.

The setup in the CoreFlow2 switch, which assumes that the loopback interface 10.10.10.10 has been already created, is as follows:

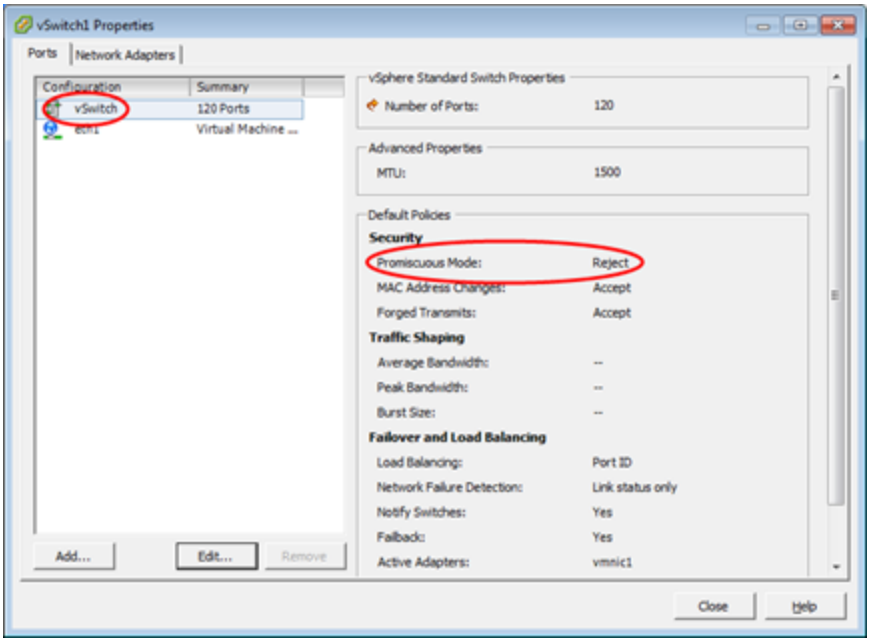
```
interface tun.0.1
  tunnel destination 10.10.10.1
  tunnel mode gre l2 ge.1.23
```

```
  tunnel mirror enable
  tunnel source 10.10.10.10
  no shutdown
  exit
interface tun.0.2
  tunnel destination 10.10.10.2
  tunnel mode gre l2 ge.1.23
  tunnel mirror enable
  tunnel source 10.10.10.10
  no shutdown
  exit
```

Additional interfaces are added for each GRE tunnel.
Note: Jumbo frames must be enabled for each interface that will be passing the GRE traffic.

VMWare TAP Interface

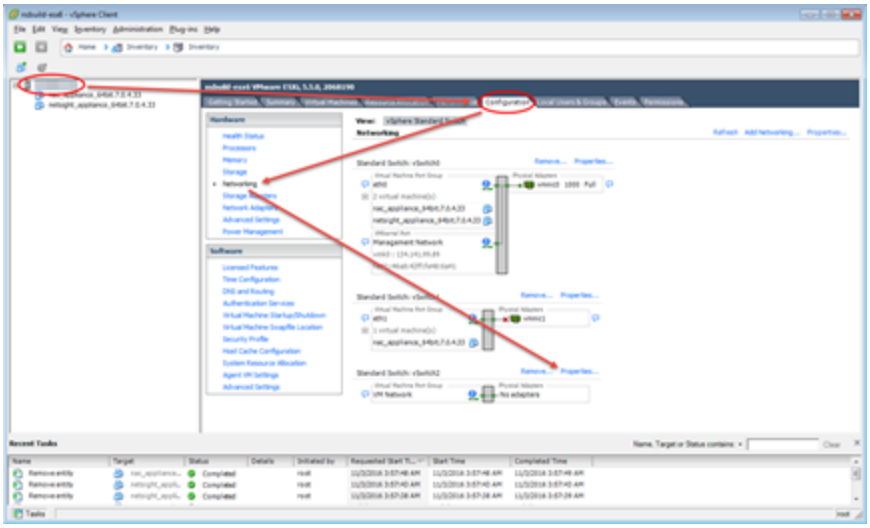
If you are using a vSwitch's TAP interface, you must enable promiscuous mode on the vSwitch to allow the ExtremeAnanlytics engine to capture packets. Promiscuous mode, which is typically used for packet sniffing, will allow the virtual ExtremeAnalytics engine to see all of the mirrored traffic. By default, promiscuous mode for a newly created vSwitch is disabled (that is, set to Reject) as shown in the following figure.



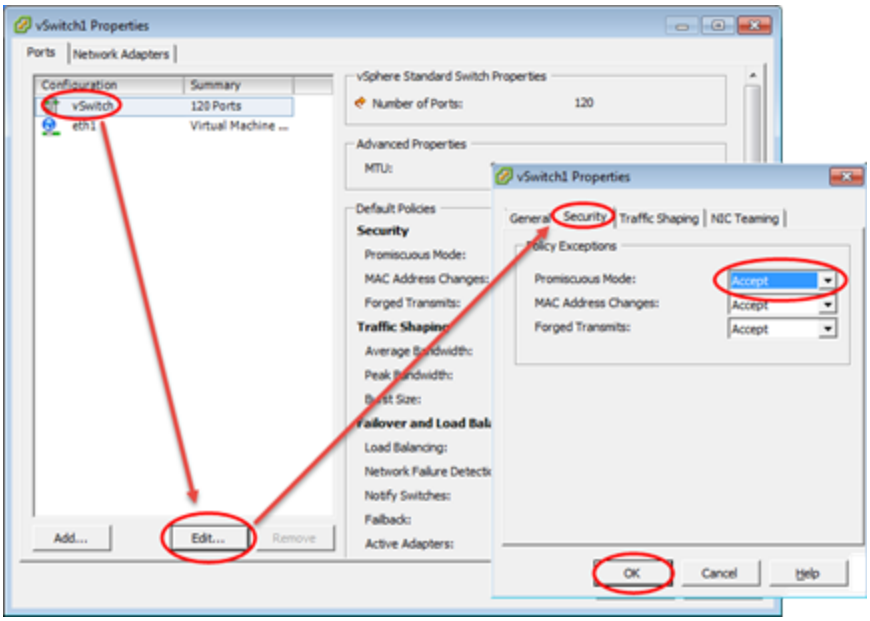
vSwitch Properties

To enable promiscous mode on the vSwitch:

1. Select the desired ESXi host from the device tree in the left-hand panel.
2. Click the Configuration tab.
3. Click Properties to the right of the vSwitch that is used by the ExtremeAnanlytics monitor interface.



4. Select the appropriate vSwitch on the Ports tab of the vSwitch Properties window.
5. Click Edit.
6. Select the Security tab.
7. Change the Promiscuous Mode drop-down list value from Reject to Accept.
8. Click OK.



Promiscuous mode is now enabled for the newly created port group TAP interface. On the Ports tab, the Promiscuous Mode setting for the appropriate vSwitch should indicate that promiscuous mode is now set to Accept (that is, enabled).

Troubleshooting

Use the following section to troubleshoot the following deployment issues:

- Network Connectivity
- Unidentified Entries in ExtremeAnalytics

Network Connectivity

Follow the procedure in this section to troubleshoot issues with network connectivity in your ExtremeAnanlytics deployment:

1. After configuring NetFlow on an Extreme CoreFlow2 switch, verify that NetFlow packets are being forwarded from the switch with the following command:

   ```
   show NetFlow statistics
   ```

2. Ensure that ExtremeCloud IQ - Site Engine can ping the ExtremeAnalytics engine's eth0 interface.
3. Check the status of the ExtremeAnalytics engine in ExtremeCloud IQ - Site Engine. If the ExtremeAnalytics engine's status is not green in ExtremeCloud IQ - Site Engine check the SNMPv3 configuration in the engine and in ExtremeCloud IQ - Site Engine.
4. Check the GRE tunnels, if used. The GRE interfaces should be listed and their status should be UP.

   ```
   ifconfig
   eth0      Link encap:Ethernet  HWaddr 00:50:56:8d:2c:22
             inet addr:192.168.30.136  Bcast:192.168.30.255
   Mask:255.255.255.0
             inet6 addr: fe80::250:56ff:fe8d:2c22/64
   Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500
   Metric:1
             RX packets:233517 errors:0 dropped:0 overruns:0
   frame:0
             TX packets:32157 errors:0 dropped:0 overruns:0
   carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:52578402 (52.5 MB)  TX bytes:23198996
   ```

```
(23.1 MB)

eth1      Link encap:Ethernet  HWaddr 00:50:56:8d:2c:23
          inet6 addr: fe80::250:56ff:fe8d:2c23/64
Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500
 Metric:1
          RX packets:21708377 errors:0 dropped:0
overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14737338317 (14.7 GB)  TX bytes:468
(468.0 B)

gre1      Link encap:Ethernet  HWaddr d6:05:a1:a0:21:82
          inet6 addr: fe80::d405:a1ff:fea0:2182/64
Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1462
Metric:1
          RX packets:198332 errors:0 dropped:193599
overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38016139 (38.0 MB)  TX bytes:468 (468.0
B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2199776 errors:0 dropped:0 overruns:0
frame:0
          TX packets:2199776 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:0
          RX bytes:780936519 (780.9 MB)  TX
bytes:780936519 (780.9 MB)
```

5. Verify that GRE traffic is received on the interface on all interfaces carrying GRE traffic. If GRE traffic is received, the decoded packets contain the protocol GREv0.

```
tcpdump -i eth0 ip proto 47

tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture
size 65535 bytes
11:20:33.462765 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 208: STP Unknown STP protocol (0x04)
11:20:33.809119 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:33.809307 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:35.462599 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 208: STP Unknown STP protocol (0x04)
11:20:35.578897 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:35.578942 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:37.443832 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 350: IP 0.0.0.0.bootpc >
255.255.255.255.bootps: BOOTP/DHCP, Request from
00:50:56:8d:b4:63 (oui Unknown), length 300
11:20:37.462472 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 208: STP Unknown STP protocol (0x04)
11:20:37.508985 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:37.509044 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:39.218957 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
```

```
11:20:39.219410 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:39.462607 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 208: STP Unknown STP protocol (0x04)
11:20:40.829045 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:40.829220 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:41.462537 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 208: STP Unknown STP protocol (0x04)
11:20:41.999060 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 200: IS-IS, L1 CSNP, src-id
20b3.9992.b666.00, length 179
11:20:41.999072 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 200: IS-IS, L1 CSNP, src-id
20b3.9992.b666.00, length 179
11:20:42.579115 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:20:42.579138 IP 192.168.10.11 > AppID51.demo.com:
GREv0, length 235: IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
```

6. Verify that traffic is received through the GRE tunnel.

```
Tcpdump –i gre1
root@AppID51.demo.com:~$ tcpdump -i gre1
tcpdump: WARNING: gre1: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on gre1, link-type EN10MB (Ethernet), capture
size 65535 bytes
11:26:03.250600 IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:26:03.250768 IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:26:03.465581 STP Unknown STP protocol (0x04)
11:26:05.030617 IS-IS, p2p IIH, src-id 20b3.9992.b666,
```

```
length 214
11:26:05.030625 IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:26:05.465612 STP Unknown STP protocol (0x04)
11:26:06.700610 IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:26:06.700706 IS-IS, p2p IIH, src-id 20b3.9992.b666,
length 214
11:26:07.465432 STP Unknown STP protocol (0x04)
9 packets captured
9 packets received by filter
0 packets dropped by kernel
```

7. Ensure the ExtremeAnalytics engine is receiving NetFlow records.
    1. Sniff the interface containing NetFlow records. In a configuration that does not use GRE tunnels, NetFlow exists on the management interface (eth0) for NetFlow traffic.

       ```
       $ tcpdump -i eth0 udp port 2055
       ```

       Note: Press [CTRL]+[C] to interrupt sniffing.
    2. Observe the NetFlow frames, if any are returned. The time it takes for NetFlow frames to display may vary depending on the amount of traffic monitored, which is typically twice the export-interval value. In the conventional configuration, NetFlow frames are returned in two minutes.

       If there are NetFlow frames, proceed to Step 9.

       If there are NO NetFlow frames, proceed to Step 7c.
    3. Access the CLI on the CoreFlow2 switch and enter the following command:

       ```
       show config netflow
       ```
    4. There should be an entry for an export-destination for the IP address on the ExtremeAnalytics engine associated with the interface expected to receive NetFlow traffic.

       If the entry exists, ping the address of the ExtremeAnalytics interface expected to receive NetFlow. In instances where asymmetric routing occurs, it may be necessary to view these frames through tcpdump on the ExtremeAnalytics engine.

If not, enter the following line into the CoreFlow2 switch CLI:

```
set NetFlow export-destination (IP address) 2055
```

8. Verify that the tunnel source switches are reachable from the ExtremeAnalytics engine interfaces.
   In this example, the tunnel is established between 192.168.10.11 and the ExtremeAnalytics engine 192.168.30.136.

```
root@AppID51.demo.com:~$ ping 192.168.10.11 -I
192.168.30.136
PING 192.168.10.11 (192.168.10.11) from 192.168.30.136 :
56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_req=1 ttl=63 time=2.24
ms
64 bytes from 192.168.10.11: icmp_req=2 ttl=63 time=2.01
ms
64 bytes from 192.168.10.11: icmp_req=3 ttl=63 time=1.61
ms
--- 192.168.10.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
2002ms
rtt min/avg/max/mdev = 1.618/1.958/2.248/0.264 ms
```

9. Ensure the ExtremeAnalytics Management Upstart service is running:

   ```
   $ appidctl status
   ```

   The status command displays the state of the two processes running on the ExtremeAnalytics engine. Both services should be in a state of start/running and have a process ID associated with them.

   If the processes are listed with state of start/running, proceed to Step 11.

   If the processes have a state other than start/running, proceed to Step 10.

10. Check the state:
    1. View the contents of the `/var/log/appidmgmtserver.log` file. If there are any stack traces or notable errors, forward the content of the file to Extreme Support.
    2. Restart the service.

       ```
       $ appidctl restart
       ```

3. Wait 30 seconds, then check the status of the processes again.

   ```
   $ appidctl status
   ```

   Assuming the processes are listed with state of start/running, proceed to Step 11.

11. Ensure the Java process is running.

    ```
    $ ps -eaf | grep java
    ```

12. Ensure the ExtremeAnalytics management process is listening on NetFlow 2055.

    ```
    $ netstat -pan | grep java
    ```

13. Ensure that reverse path check is disabled. Some ExtremeAnalytics multi-interface configurations discard packets if reverse path check is enabled. Reverse path check should be disabled by the interface configuration script.

    ```
    root@AppID51.demo.com:~$ sysctl -a | grep -F '.rp_filter'
    net.ipv4.conf.all.rp_filter = 1
    net.ipv4.conf.default.rp_filter = 1
    net.ipv4.conf.eth0.rp_filter = 1
    net.ipv4.conf.eth1.rp_filter = 0
    net.ipv4.conf.gre0.rp_filter = 1
    net.ipv4.conf.gre1.rp_filter = 1
    net.ipv4.conf.lo.rp_filter = 1
    ```

    A value of 0 means reverse path check is disabled for that interface.

Unidentified Entries in ExtremeAnalytics

If entries are displayed, but unidentified in the ExtremeAnalytics Application Flows view, perform the following steps:

1. Ensure the monitoring interface contains what appears to be mirrored traffic. On the ExtremeAnalytics engine, execute the following command on the monitoring interface:

   ```
   tcpdump -i eth1 -s 0 -nn -c 100
   ```

   This will capture, at most, 100 frames. If the process does not terminate quickly, press [CTRL]+[C] to exit the tcpdump capture process.

If there are frames that appear to be the monitored traffic, proceed to Step 5.

If there are no frames, proceed to Step 2.

2. If no frames appear to be the monitored traffic, ensure the Java process is running:

   ```
   $ ps -eaf | grep java
   ```

3. Ensure the ExtremeAnalytics management process is listening on NetFlow 2055:
   ```
   $ netstat -pan | grep java
   ```

4. If no frames appear to be representative of the traffic being mirrored, examine the CoreFlow2 switch configuration.

   1. On the CoreFlow2 switch, execute the following commands:

      ```
      show config policy
      show config mirror
      show running-config (If GRE is enabled)
      ```

   2. Ensure the policy profile configuration contains the port for the traffic mirror.
   3. Ensure the mirror configuration contains the port for the mirror interface connection.
   4. Verify the GRE tunnel configuration is correct. Execute the following commands:

      ```
      config
      show tunnel
      ```

      The affected interface should have an Admin status of Enabled and an Oper status of Up.

   5. Verify the port statuses of connections to the CoreFlow2 switch. Execute the following command:

      ```
      show port status -interesting
      ```

      This command will only list ports with an Oper status of Up. Ensure relevant connections are Up.

   6. View packet counters on the mirror feed:

      ```
      show port counters (mirror feed interface)
      ```

Execute this command a few times and note the changes in the interface and switch counters. If there are no changes or few changes, the external port mirror configuration may be incorrect or there may be no traffic mirrored.

5. Ensure IPFIX records are being generated. On the ExtremeAnalytics engine, execute the following command on the loopback interface (use [CTRL]+[C] to terminate):

   ```
   tcpdump -i lo udp port 9191
   ```

   If IPFIX records exist, proceed to Step 8.

   If IPFIX records do not exist, proceed to Step 6.

6. Ensure the ExtremeAnalytics service is running.

   ```
   $ appidctl status
   ```

   Execute this command multiple times sequentially to ensure the process is not restarting continually.

7. Navigate to /opt/appid/conf/appidconfig.xml and confirm the *<interface>* tag is designated to the correct interface. Also confirm the IPFIX host destination is itself: 127.0.0.1 with port 9191.

8. Ensure the management process is listening on NetFlow 9191:

   ```
   $ netstat -pan | grep java
   ```

   An entry should indicate that the ExtremeAnalytics management process is listening on all interfaces on UDP 9191.